

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ГОРЯСІКОРСЬКОГО»
Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Завідувач кафедри

_____ Олександр Коваль

«___» _____ 2020 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інформаційні технології моніторингу довкілля»

спеціальності 122 «Комп'ютерні науки та інформаційні технології»

на тему: «Адміністрування системи підготовки та атестації науково – педагогічних кадрів вищої кваліфікації в аспірантурі університету»

Виконав:

студент IV курсу, групи ТМ-61

Строган Роман Сергійович

Керівник:

професор, д. т. н.

Гаврилко Євген Володимирович

Рецензент:

Професор д. т. н.

Барабаш Олег Володимирович

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ – 2020 року

Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 122 Комп'ютерні науки та інформаційні технології

Спеціалізація Інформаційні технології моніторингу довкілля

АТВЕРДЖУЮ

Завідувач кафедри

Олександр Коваль

(підпис)

” ” 2020р.

ЗАВДАННЯ

на дипломну роботу студенту

Строгану Роману Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи «Адміністрування системи підготовки та атестації науково – педагогічних кадрів вищої кваліфікації в аспірантурі університету»

керівник роботи Гаврилко Євген Володимирович, д. т. н., професор

(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ”25” травня 2020р. № **1168-с**

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи

мова програмування JavaScript, програмна платформа Node.js, середовище розробки JetBrains WebStorm

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

проаналізувати проблему організації вступної компанії до аспірантури університету за освітніми програмами доктора філософії, розробити систему супроводження та аналізу вступу до аспірантури університету.

5. Перелік ілюстративного матеріалу

схеми архітектури програмного продукту, знімки інтерфейсу, знімки структури проекту

7. Дата видачі завдання ” ” 201__ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи		
2.	Вивчення та аналіз задачі		
3.	Розробка архітектури та загальної структури системи		
4.	Розробка структур окремих підсистем		
5.	Програмна реалізація системи		
6.	Оформлення пояснювальної записки		
7.	Захист програмного продукту		
8.	Передзахист		
9.	Захист		

Студент

(підпис)

Р.С Строган.

(прізвище та ініціали,)

Керівник роботи

(підпис)

Є.В Гаврилко.

(прізвище та ініціали,)

АНОТАЦІЯ

Дипломна робота складається зі вступу, чотирьох розділів, висновку, перелику посилань з 5 найменувань, і містить 28 рисунків. Повний обсяг дипломної роботи складає 52 сторінки, з яких перелік посилань складає 1 сторінку.

Актуальність теми. Проблема покращення та автоматизації якості процесу вступу наявна з року в рік. Наразі існують автоматизовані системи, які спрощують та покращують процес подання заяв вступниками, а також дозволяють моніторинг змін рейтингових списків. Наявні системи забезпечують покриття лише першого та другого освітніх рівнів вищої освіти, проте, існує також третій рівень освіти. Наразі, для третього рівня відсутні автоматизовані системи вступу. Це зумовлює актуальність та необхідність розробки програмного засобу автоматизації, тому що автоматизація процесу дозволяє економити ресурси, швидше подавати та перевіряти заявки вступників, зменшує ризик помилок, що залежать від людського фактору.

Мета дослідження. Метою дослідження є розробка програмного засобу для супроводження та аналізу вступних заяв, виявлення нових підходів та принципів проведення вступної кампанії на третій рівень вищої освіти.

Для вирішення поставленої задачі було сформовано **завдання дослідження**, які визначають напрям дослідження та його структуру:

- аналіз існуючого програмного забезпечення;
- визначення основних потреб за вимог до автоматизованого програмного продукту;
- аналіз процесу вступу на третій освітній рівень вищої освіти;
- аналіз можливості удосконалення існуючих процесів вступної кампанії;
- розробка програмного продукту для автоматизації, аналізу та супроводження вступу спеціалістів до третього рівня вищої освіти.

Об'єктом дослідження є комп'ютерні інформаційні системи і технології.

Предметом дослідження є комп'ютерні інформаційні системи з

використанням новітніх технологій розробки програмного забезпечення.

Практичне значення отриманого результату. Практичним значенням дипломної роботи є розробка системи супроводження, моніторингу та аналізу вступу на на навчання на третій рівень вищої освіти – доктора філософії.

ABSTRACT

Thesis consists of an introduction, 4 sections, conclusion, a list of references with 5 titles, and contains 28 pictures. The full volume of the diploma work is 52 pages, and list of references consist 1 page.

Actuality of theme. The problem of improving and automating the quality of the accession process is present from year to year. Currently, there are automated systems that simplify and improve the application process for entrants, as well as allow monitoring of changes in the ranking lists. The existing systems cover only the first and second levels of higher education, however, there is also a third level of education. Currently, there are no automated entry systems for the third level. This determines the relevance and needs to develop automation software because process automation saves resources, faster submission, and verification of applications reduces the risk of errors that depend on the human factor.

The aim of the study. The purpose of the study is to develop a software tool for support and analysis of admissions, identifying new approaches and principles of the admission campaign to the third level of higher education.

To solve this problem, a research task was formed, which determine the direction of the study and its structure:

- analysis of existing software;
- identification of basic needs for the requirements of the automated software product;
- analysis of the process of admission to the third educational level of higher education;
- analysis of the possibility of improving the existing processes of the introductory campaign;
- development of a software product for automation, analysis, and support of specialists entry to the third level of higher education.

The object of the research is computer information systems and technologies.

The subject of the research is computer information systems using the latest software development technologies.

The practical significance of the result. The practical significance of the thesis is the development of a system of support, monitoring, and analysis of the entry of specialists to the third level of higher education

ЗМІСТ

Перелік умовних позначень, символів, скорочень і термінів.....	10
Вступ.....	11
1 Задача розробки програмної системи супроводження, моніторингу та аналізу вступу спеціалістів до третього рівня освіти.....	12
1.1 Характеристика програмної системи.....	13
1.2 Висновки до розділу.....	17
2 Опис методів реалізації програмної системи.....	18
2.1 Обріна методи реалізації.....	18
2.1.1 Клієнтська частина програмної системи.....	18
2.1.1.1 React.js.....	18
2.1.1.2 HTML.....	20
2.1.1.3 CSS.....	20
2.1.1.4 SASS.....	21
2.1.1.5 JavaScript.....	21
2.1.1.6 Redux.....	22
2.1.2 Серверна частина програмної системи.....	22
2.1.2.1 Node.js.....	23
2.1.2.2 MongoDB.....	24
2.2 Висновки до розділу.....	25
3 Опис програмного забезпечення програмної системи.....	26
3.1 Опис програмної реалізації програмної системи.....	26
3.2 Опис програмної реалізації серверної частини.....	27
3.2.1 REST.....	27
3.2.2 Структура серверної частини.....	28
3.2.3 Сервіс взаємодії з базою даних.....	29
3.2.4 Сервіс реєстрації та авторизації.....	29
3.3 Опис програмної реалізації клієнтської частини.....	31

3.3.1 Реактивне програмування.....	33
3.3.2 Взаємодія з серверною частиною.....	34
3.3.3 Сторінка аутентифікації або реєстрації користувача.....	35
3.3.4 Сторінка подання реєстраційної форми	36
4 Інтерфейс взаємодії та робота користувача з системою.....	38
4.1 Інтерфейс системи.....	38
4.1.1 Функціональні можливості користувача.....	40
4.1.2 Функціональні можливості адміністратора.....	45
4.2 Висновки до розділу.....	50
Висновки.....	52
Список використаних джерел.....	53

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ОС – операційна система.

JavaScript – мова програмування.

HTML – мова розмітки веб-сторінки.

CSS – каскадна таблиця стилів.

JetBrainsWebStorm - середовище розробки.

SPA – Single Page Application—односторінковий додаток.

Фреймворк – програмна платформа, визначаюча структуру програмної системи.

Маршрутизація – визначення маршрутів додатку.

ВСТУП

Для розв'язку проблеми супроводження, моніторингу та аналізу вступу до третього рівня вищої освіти було розроблено програмну систему, що дозволяє проводити прийом заяв на навчання, переглядати рейтингову інформацію вступника, а також відображати адміністративну частину, за допомогою якої можна аналізувати заяви вступників, та обробляти їх у визначеному порядку.

Програмна система призначена для супроводження, моніторингу та аналізу вступу спеціалістів до третього рівня вищої освіти. Програмна система ділиться на дві підсистеми, одна з яких призначена для користувача (вступника), а друга – для адміністратора.

Перша підсистема дозволяє спеціалістам надавати дані для вступу до третього рівня вищої освіти – аспірантури – у електронному вигляді, слідкувати за процесом вступу. Після зарахування, аспірант має можливість переглядати дані, що стосуються його навчального процесу (відмітки присутності, оцінки, статистику), а також загальні дані (рейтинговий список, тощо).

Друга підсистема надає можливості моніторингу, аналізу та контролю процесу вступу адміністратором, такі як прийом та редагування вхідних заяв, внесення залікових балів.

Створена програмна система є спрощенням процесу вступу, може автоматизувати ручну роботу, покращує зручність аналізу даних, полегшує супроводження учасників вступної кампанії.

У першому розділі пояснювальної записки сформовано мету роботи, описано постановку задачі.

У другому розділі розроблено та описано прототип створеної програмної системи.

У третьому розділі надано опис методів реалізації програмної системи.

У четвертому розділі описується програмна реалізація створеного програмного продукту.

У п'ятому розділі надана інструкція користувача для роботи з системою, описано інтерфейс взаємодії з системою.

1 ЗАДАЧА РОЗРОБКИ ПРОГРАМНОЇ СИСТЕМИ СУПРОВОДЖЕННЯ, МОНІТОРИНГУ ТА АНАЛІЗУ ВСТУПУ СПЕЦІАЛІСТІВ ДО ТРЕТЬОГО РІВНЯ ВИЩОЇ ОСВІТИ

Метою дипломної роботи є розробка програмної системи у якості веб-додатку, який буде надавати можливість супроводження, моніторингу та аналізу вступної кампанії спеціалістів до третього рівня вищої освіти.

Призначенням створеного програмного продукту є надання інтуїтивно зрозумілого інтерфейсу взаємодії для учасників вступної кампанії та адміністратора. При розробці програмної системи було виконано наступні кроки та розв'язано наступні задачі:

1. аналіз подібних наявних програмних систем;
2. створено технічне завдання для програмної системи;
3. спроектовано структуру системи;
4. спроектовано архітектуру програмного продукту;
5. розроблено програмну систему;
6. протестовано розроблену програмну систему;

Перелік необхідного функціоналу, який повинна забезпечувати програмна система:

1. можливість подання вступником вхідних даних;
2. автоматичний розрахунок рейтингового списку вступників;
3. супроводження вступної кампанії адміністратором;
4. моніторинг подачі заяв адміністратором;
5. аналіз вступної кампанії адміністратором;
6. редагування вхідних даних адміністратором.

1.1 Характеристика програмної системи

Перелік основного функціоналу створеної програмної системи:

1. функція подачі вхідних даних для участі у вступній кампанії;
2. функція моніторингу вступником та адміністратором результатів вступної кампанії вступника;
3. перегляд результатів іспитів;
4. перегляд рейтингового списку;
5. функція редагування вхідних даних вступника;
6. функція створення груп;
7. функція встановлення оцінок;
8. функція встановлення додаткових балів, які можуть вплинути на рейтинг вступника;
9. функція фільтрації вступників за певними критеріями;
10. функція перегляду статистики вступної кампанії.

Інтерфейс взаємодії користувача передбачає головну сторінку, з якої можна перейти на сторінку реєстраційної форми або сторінку рейтингового списку.

Сторінка подання реєстраційної форми передбачає введення таких даних:

- прізвище;
- ім'я;
- по батькові;
- стать;
- дата народження;
- спеціальність – випадаючий список;
- факультет – випадаючий список;
- кафедра – випадаючий список;
- форма навчання – випадаючий список;
- форма оплати – випадаючий список;
- іноземна мова – випадаючий список;

- ВНЗ, що закінчив вступник;
- рік закінчення ВНЗ;
- освітній ступінь – випадючий список;
- наявність відзнаки;
- електронна пошта;
- пароль;
- потреба в гуртожитку;
- середній бал диплому;
- кількість наукових публікацій;
- прізвище наукового керівника;
- додатковий коментар;

Після подання вступником заяви, даня зберігаються у базі даних, після чого адміністратор може обробити заяву вступника.

Інтерфейс взаємодії адміністратора передбачає наступні розділи:

- перегляд поданих вступниками заяв;
- пошук конкретної заяви за вхідними даними;
- деталізована інформація про кожну заяву;
- підтвердження заяви абітурієнта;
- відхилення заяви абітурієнта;
- перегляд списку зарахованих вступників;
- перегляд статистики вступної кампанії;

На рисунку 1.1 Зображено діаграму прецедентів системи.



Рисунок 1.1 Діаграма прецедентів системи

На діаграмі послідовності програмної системи для ролі абітурієнта (рисунк 1.2) схематично описана взаємодія користувача з елементами програмної системи. Абітурієнт має можливість введення вхідних даних для подання реєстраційної форми. Дані з реєстраційної форми зберігаються в базі даних. Абітурієнт має змогу побачити свій рейтинговий бал за допомогою пошукової форми рейтингового списку.

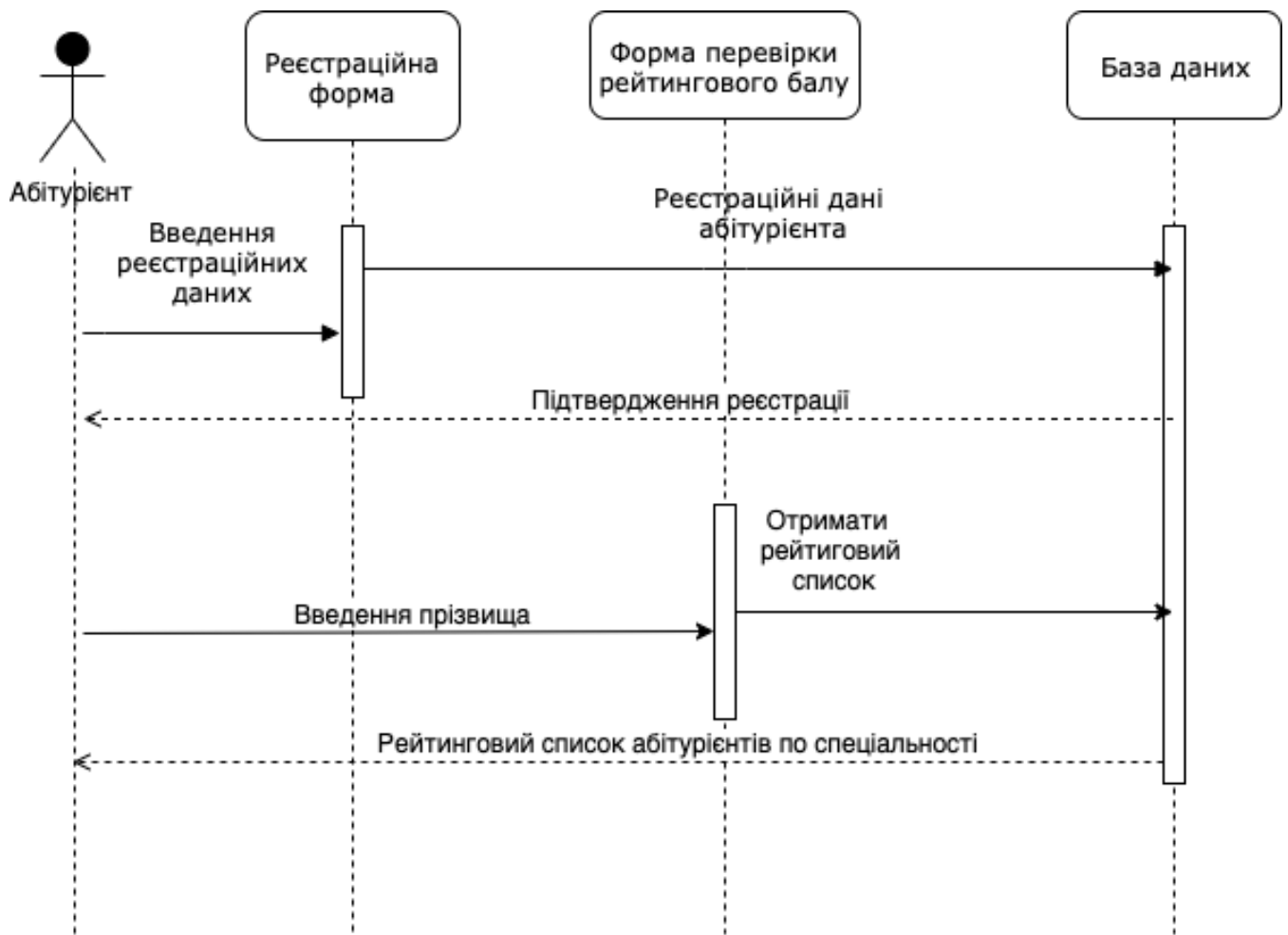


Рисунок 1.2 Діаграма послідовності системи для ролі Абітурієнт

На діаграмі послідовності програмної системи (рисунок 1.3) схематично зображено взаємодію користувача з елементами супроводження вступу спеціалістів до третього рівня вищої освіти. Адміністратор має змогу пройти авторизацію в програмній системі за допомогою пари логін-пароль. В разі успішного входу адміністратор може переглядати список заяв вступників, підтверджувати або відмінювати їх реєстрацію, переглянути список підтверджених абітурієнтів, вносити бали за екзамени та робити зміни у наявних даних абітурієнтів. Всі зміни автоматично враховуються та відображаються у рейтинговому списку. Адміністратор може зарахувати вступника до університету, надати йому номер групи.

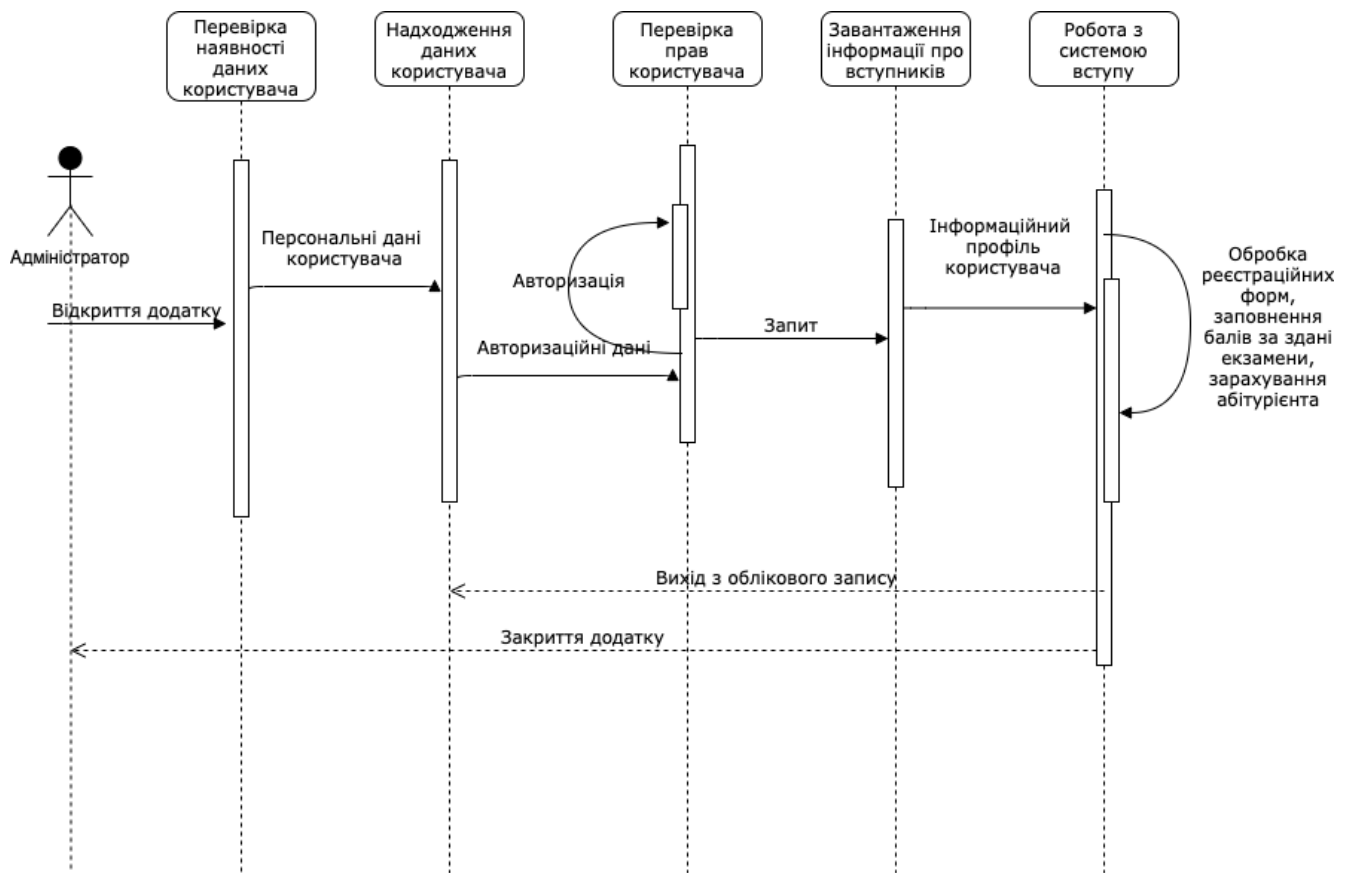


Рисунок 1.3 Діаграма послідовності системи для ролі адміністратора

1.2 Висновки до розділу

В даному розділі описано задачу, результатом вирішення якої є створення програмної системи для супроводження, моніторингу та аналізу вступної кампанії до третього рівня вищої освіти – аспірантури.

Система ділиться на дві підсистеми:

- підсистема для вступників;
- підсистема для адміністратора.

Вступник може подавати реєстраційну форму та переглядати рейтингові списки.

Адміністратор може приймати рішення щодо підтвердження або скасування заяви вступника, аналізувати статистику вступної кампанії.

2 ОПИС МЕТОДІВ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ

Під час підготовки до розробки будь-якого програмного продукту обов'язковим є правильний вибір засобів програмної реалізації, технологій, патернів проектування.

Розробка системи проводилась в середовищі JetBrainsWebStorm.

2.1 Обрані методи реалізації

Для реалізації програмної системи було вирішено створити веб-додаток задля досягнення кросплатформенності. Тобто, користуватись додатком можна на будь-якій платформі, якщо на ній доступні веб-браузери (Windows, Linux, macOS, Android, iOS тощо).

2.1.1 Клієнтська частина програмної системи

Клієнтською частиною програмного продукту є та, що запускається за виконується на комп'ютері користувача. В даному випадку, доступ до програмної системи можна отримати за допомогою веб-браузера.

2.1.1.1 React.js

React.js це бібліотека, написана мовою програмування JavaScript. Вона створена для побудови інтерфейсів користувача.

React надає такі можливості:

- двостороннє зв'язування;
- маршрутизація;
- Shadow DOM;

- компонентний підхід.

ShadowDOM – Тіньовий DOM– використовується для інкапсуляції. Завдяки йому, кожен компонент має власне, тіньове, DOM-дерево, до якого не можна звернутись з головного документа.

React дозволяє будувати справді великі та масштабуємі проекти. Так як вся логіка компонентів описується за допомогою JavaScript, можна з легкістю передавати складні дані та зберігати стан додатку окремо від DOM.

Компоненти можуть використовувати сервіси, які можуть забезпечувати особливі функції, що не пов'язані безпосередньо з представленням.

Для швидкого розгортання React-додатку можна використовувати **CreateReactApp**, який дозволяє «з коробки» почати розробку та не замислюватися над налаштуваннями збірника, транспілятора тощо.

CreateReactApp потребує встановленої Node.js версії більше 8.10 та npm версії більше 5.6. Для розгортання додатку необхідно в інтерфейсі командної строки ввести команду **npm create-react-app my-app**, після чого застосунок самостійно згенерує структуру проекту та проведе всі необхідні налаштування.

На рисунку 2.1.1 можна побачити схему роботи та рендерингу компонентів React-додатка.

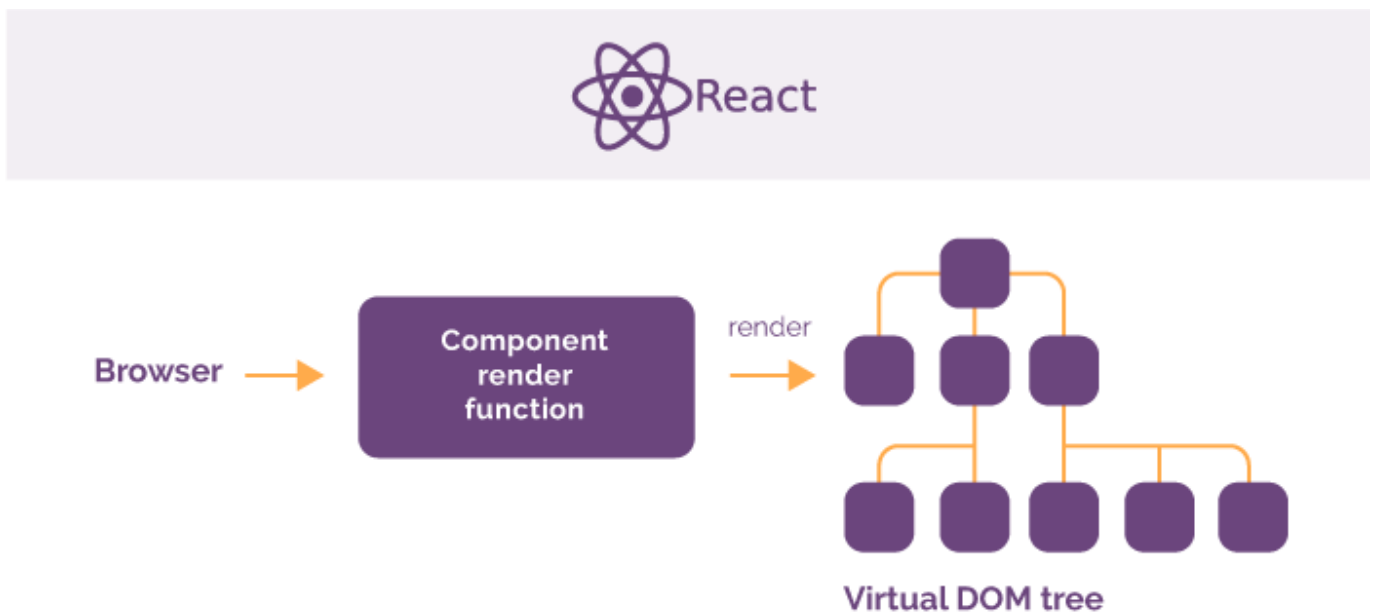


Рисунок 2.1.1 Схема рендерингу React-компонентів

2.1.1.2 HTML

HTML – мова гіпертекстової розмітки, яка використовується для створення веб-сторінок і наразі, фактично, є єдиним інструментом для розмітки документів.

HTML описує структуру веб-сторінки семантично, тобто кожен блок має власну сутність.

HTML-елементи це засоби для створення структури документа та можуть описувати структурну семантику для текстів, заголовків, списків тощо. HTML-елементи ще називаються HTML-тегами.

Хоча й HTML при своєму створенні не планувався для тих цілей, для яких він використовується зараз, остання його версія, HTML5, відкриває дуже багато мультимедійних можливостей, таких як:

- програвання відеозаписів;
- програвання аудіозаписів;
- малювання за допомогою Canvas;
- Drag'n'Drop.

2.1.1.3 CSS

CSS – Cascading style sheets – мова для опису зовнішнього вигляду веб-сторінок. За допомогою цієї мови можна стилізувати розміри, кольори, шрифти, поведінку при наведенні миші на елемент, тощо. За допомогою CSS здійснюється адаптація зовнішнього вигляду веб-сторінки для відображення на екранах різного розміру.

CSS не залежить від HTML та може використовуватись з мовою будь-якої розмітки на основі XML.

Веб-браузер застосовує правила до документа, щоб впливати на його зовнішній вигляд. Правила формуються за допомогою набору властивостей (для впливу на ширину, колір, позиціонування тощо), а також селекторів, які дають браузеру можливість зрозуміти, якого елемента стосуються дані властивості.

2.1.1.4 SASS/SCSS

SASS (Syntactically awesome style sheets) – CSS-препроцесор, який транспілюється в CSS-код. Препроцесори призначені для полегшення написання коду, уникнення дублювання коду. SASS має можливість введення змінних и також «домішок» («mixins»). Це полегшує підтримку цілісності даних.

2.1.1.5 JavaScript

JavaScript – динамічна, мультипарадигмальна мова програмування. Є реалізацією стандарту ECMAScript. Наразі фактично є єдиною мовою для створення сценаріїв веб-сторінок.

Також, JavaScript використовується для:

- створення SPA – React, Angular, Vue;
- серверне програмування – Node.js;
- створення стаціонарних застосунків – Electron, NW.js;
- створення мобільних застосунків – ReactNative, Cordova;

JavaScript – мультипарадигмальна мова програмування. Він має низку властивостей об'єктно-орієнтовної мови програмування, проте завдяки прототипній концепції підтримка об'єктів в ній відрізняється від традиційних мов ООП.

Крім того, JavaScript має низку властивостей, що притаманні функціональним мовам, серед яких - об'єкти як списки, каррінг, анонімні функції, замикання, функції як об'єкти першого пласу – що додає мові додаткову гнучкість.

JavaScript містить декілька вбудованих об'єктів: Global, Object, Error, Function, Array, String, Boolean, Number, Math, Date, RegExp. Крім того, містить набір вбудованих функцій, які не обов'язково є функціями або методами, а також набір вбудованих операторів, що правляють логікою виконання програм.

Синтаксис JavaScript є успадкованим від C, але спрощений порівняно з ним задля того, щоб зробити мову легкою до вивчення.

2.1.1.6 Redux

Redux - відкрита JavaScriptбібліотека, призначена для керування станом програми. Найбільш поширене використання – у зв'язці з Reactдля побудови інтерфейсів взаємодії користувача.

Reduxбудується на трьох основних концептах:

- **Єдине джерело істини.** Reduxзберігає стан усього застосунку в дереві об'єктів в одному сховищі. Одне дерево об'єктів полегшує налагодження або перевірку програми; це також дозволяє зберігати стан програми в процесі розробки, для її прискорення;
- **Стан призначений виключно для читання.** Єдиним способом змінити стан є відокремлення дію, об'єкт, який описує те, що сталося. Це гарантує, що ні перегляди, ні зворотні виклики ніколи не будуть змінювати стан;
- **Зміни за допомогою чистих функцій.** Редьюсери – чисті функції, які приймають чтан та дію, і повертають наступний стан. В процесі розробки можуть будти подріблені на ще менші редьюсери.

На рисунку 2.1.2 зображено схему роботи Redux.

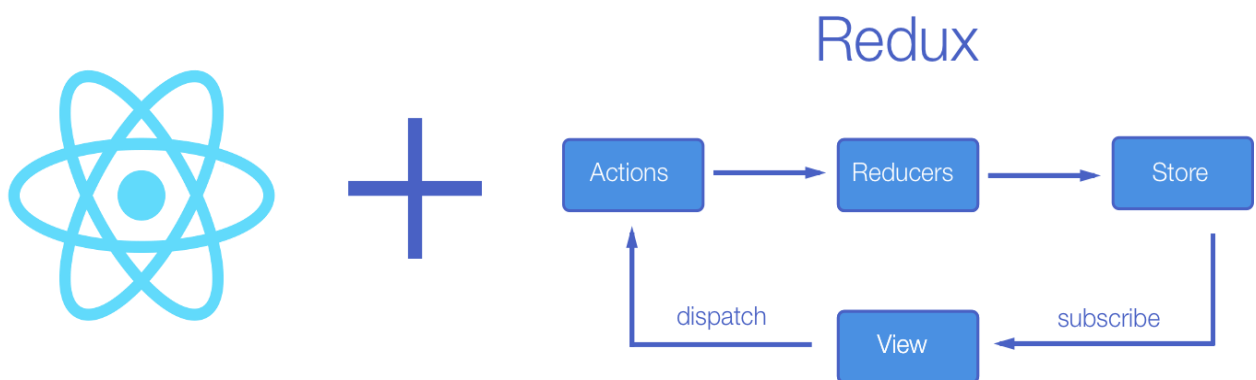


Рисунок 2.1.1 Схема роботи Redux

2.1.2 Серверна частина програмної системи

Серверна частина програмного забезпечення – та, що виконується на стороні серверу. Зазвичай, на серверній частині описана більша частина бізнес-логіки. За

допомогою серверної частини ПЗ робляться всі запити до бази даних, а також відбувається «спілкування» з клієнтом через HTTP-відповіді.

2.1.2.1 Node.js

Node.js – платформа з відкритим кодом для створення високопродуктивних мережевих застосунків, написаних мовою JavaScript.

Node.js має наступні властивості:

- асинхронна одно-нитева модель виконання запитів;
- неблокуючий ввід\вивід;
- система модулів CommonJS;
- JavaScript Chrome V8 рушій.

Для керування модулями використовується пакетний менеджер npm.

Для забезпечення обробки великої кількості паралельних запитів у Node.js використовується асинхронна модель запуску коду, заснована на обробці подій в неблокуючому режимі та визначенні обробників зворотніх викликів. Як способи мультиплексування з'єднань підтримується `epoll`, `kqueue`, `/dev/poll` і `select`. Для мультиплексування з'єднань використовується бібліотека `libuv`, для створення пулу нитей (`thread pool`) задіяна бібліотека `libeio`, для виконання DNS-запитів у неблокуючому режимі інтегрований `c-ares`. Всі системні виклики, що спричиняють блокування, виконуються всередині пулу нитей і потім, як і обробники сигналів, передають результат своєї роботи назад через неіменовані канали (`pipe`).

Для розширення функціональності застосунків на базі Node.js підготовлена велика колекція модулів, в якій можна знайти реалізацію HTTP, SMTP, XMPP, DNS, FTP, IMAP, POP3 серверів і клієнтів, модулі для інтеграції з різними веб-фреймворками, обробники WebSocket і AJAX, конектори до СКБД (MySQL, PostgreSQL, SQLite, MongoDB), шаблонізатори, CSS-рушії, реалізації криптоалгоритмів і систем авторизації (наприклад, OAuth), XML-парсери.

Стандартна збірка Node.js включає в себе набір модулів, у яких реалізовані

типові операції для взаємодії з операційною системою, файловою системою, мережею, протоколами, утиліти для обробки даних.

2.1.2.2 MongoDB

MongoDB – документо-орієнтовна система керування базами даних з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, які оперують даними у форматі ключ-значення, та реляційними СКБД, які є зручними при формуванні запитів.

MongoDB підтримує зберігання документів в JSON-подібному форматі, має гнучку мову формування запитів, може створювати унікальні індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій, парадигму Map/Reduce, реплікацію, побудову відмовостійких конфігурацій.

Основні можливості MongoDB:

- документо-орієнтовне сховище;
- досить гнучка мова формування запитів;
- повна підтримка індексів;
- профілювання запитів;
- швидкі оновлення «на місці»;
- ефективне зберігання бінарних даних великих обсягів;
- журналювання операцій;
- підтримка відмовостійкості;
- можливість масштабування;
- асинхронна реплікація;
- набір реплік, шардінг;
- підтримка парадигми Map/Reduce.

СКБД MongoDB управляє наборами JSON-подібних документів, що зберігаються в бінарному форматі BSON. Зберігання і пошук в базі даних відбувається завдяки протоколу GridFS.

2.2 Висновки до розділу

Для створення програмної системи у якості середовища розробки було обрано JetBrainsWebStorm, який надає змогу працювати з усім, що пов'язано з написанням коду. В якості технології для розробки інтерфейсу взаємодії користувача було обрано мову програмування JavaScript, фреймворк – React.js. На стороні сервера, для розробки було обрано Node.js, базу даних –MongoDB. Таке поєднання дозволяє легко підтримувати систему та масштабувати її за необхідності.

3 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОГРАМНОЇ СИСТЕМИ

Розроблена програмна система становить собою веб-додаток та серверну частину. Архітектуру програмної системи зображено на рисунку 3.1.

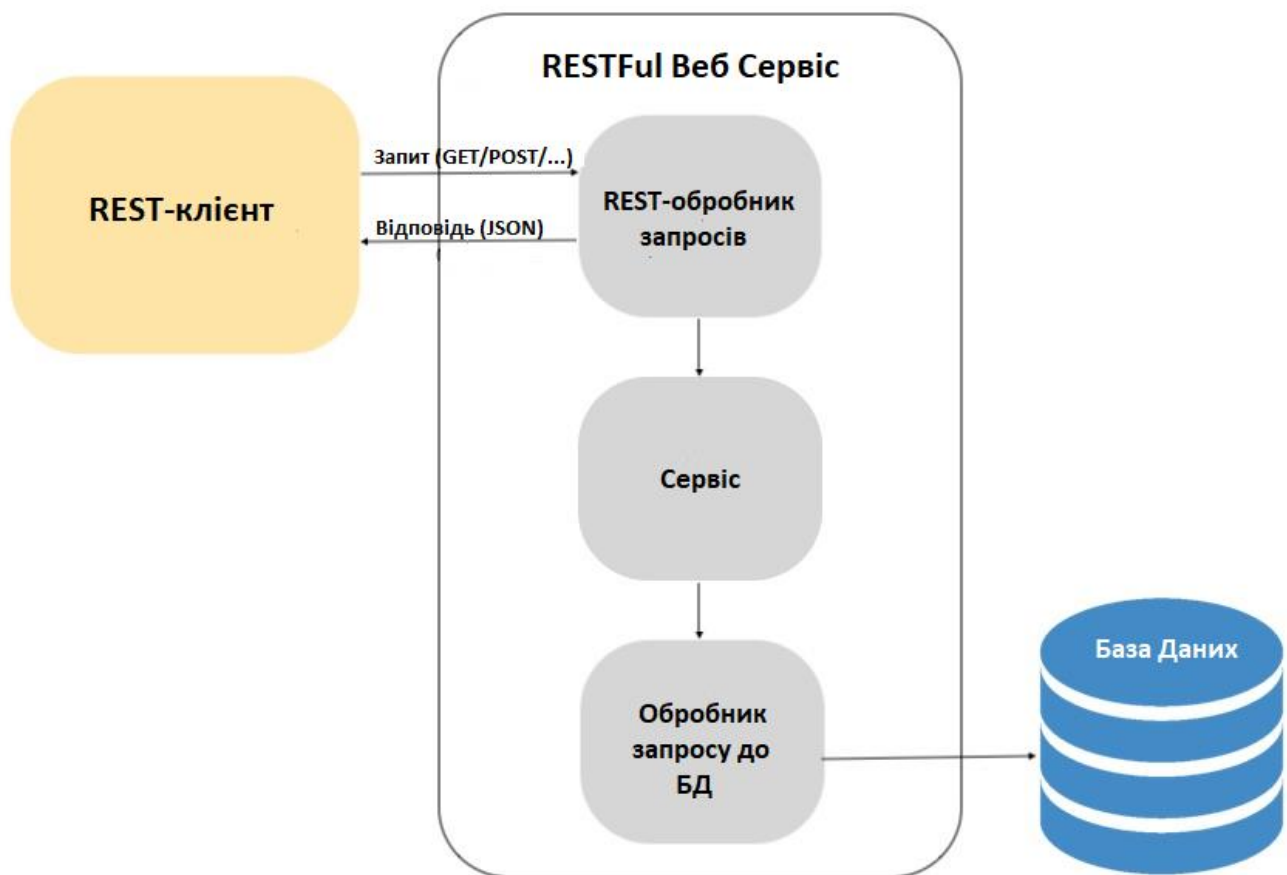


Рисунок 2.1 Архітектура програмної системи

3.1 Опис програмної реалізації програмної системи

Клієнтська частина веб-додатку створена за допомогою мови JavaScriptта фреймворку Reactчерез його швидкодію. Для реалізації серверної частини було обрано платформу Node.js, яка добре масштабується. Системою керування базою

даних було обрано MongoDB.

3.2 Опис програмної реалізації серверної частини

Серверна частина реалізована за допомогою мови програмування JavaScriptта платформи Node.js.

В ній описана основна бізнес-логіка, сервіси для реєстрації, авторизації, роботою з базою даних, моделі бази даних.

Архітектурним підходом для серверної частини програмної системи було обрано REST – RepresentativeStateTransfer.

3.2.1 REST

У RESTдані передеються у невеликій кількості стандартних форматів (HTML, XML, JSON). Будь-який REST-протокол повинен підтримувати кешування, не має залежати від мережевого прошарку, не повинен зберігати інформацію про стан між парами «запит-відповідь». Такий підхід дозволяє легко масштабувати систему.

REST має ряд архітектурних обмежень:

- **Клієнт-сервер.** Обмеження клієнт-серверної архітектури вимагає розділення відповідальності між компонентами, які займаються зберіганням та оновленням даних (сервером), та тими компонентами, які займаються представленням даних (клієнтом).
- **Відсутність стану.** Взаємодії між клієнтом і сервером не мають стану. Тобто, кожен запит повинен містити всю необхідку інформацію для його обробки, і не повинен покладатись на те, що сервер «знає» щось з минулих запитів.
- **Кешування.** Системи, написані з використанням підходу RESTповинні підтримувати кешування. Тобто, дані, що передаються з сервера, повинні містити дозвіл на кешування, та «час життя» кешу, якщо дозвіл є.

- **Однорідний інтерфейс.** Всі компоненти в архітектурі REST повинні підтримувати однорідність інтерфейсу. Це зменшує зв'язність між компонентами і дозволяє дуже просто її змінювати, за потреби.
- **Шари абстракції.** Обов'язковим є поділ на абстракції. Кожен компонент знаходиться в якомусь з шарів абстракції та може «спілкуватись» лише з компонентами під або над ним.

3.2.2 Структура серверної частини

На рисунку 3.2.1 зображену структуру серверної частини програмної системи

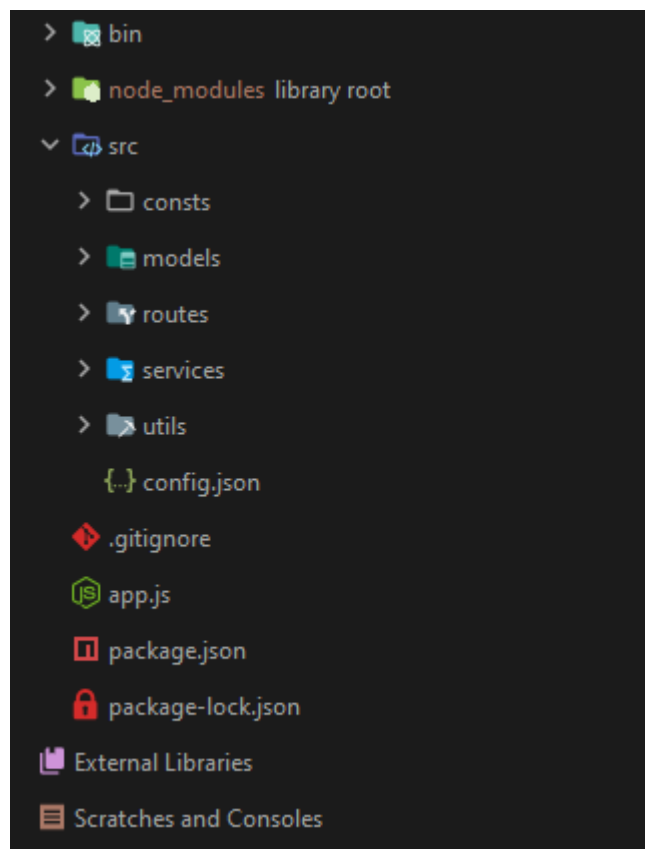


Рисунок 3.2.1 Структура серверної частини

- **bin** – папка з точкою входу програмного забезпечення серверної частини;
- **node_modules** – папка, де храняться всі бібліотеки, необхідні для роботи

серверной частини;

- **src** –папка з «самописним» кодом
- **src/consts** –папка з константами;
- **src/models** –папка з моделями бази даних;
- **src/routes** –папка з файлами, що відповідають за обробку запитів по кожному маршруту;
- **src/services** –сервіси;
- **src/utils** –засоби для специфічних задач.

3.2.3 Сервіс взаємодії з базою даних

За взаємодію з базою даних відповідає сервіс `mongooseService.js`. Він виконує всі свої функції асинхронно, завдяки чому не блокує роботу сервера на час, наприклад, пошуку інформації в базі даних. В сервісі реалізовано наступний функціонал:

- **connect** – відкриття підключення до бази даних;
- **disconnect** – закриття підключення до бази даних;
- **findDocuments** – пошук масиву записів в базі даних за заданим селектором. Повертає масив з записами або пустий масив.
- **findOneDocument** – пошук одного документу в базі даних. Повертає документ або `null`;
- **createDocument** – створити новий запис в базі даних;
- **createModel** – створити модель бази даних.

3.2.4 Сервіс реєстрації та авторизації

За реєстрацію та авторизацію відповідає сервіс `userService.js`.

При реєстрації, спочатку перевіряється унікальність користувача за допомогою даних, які він ввів до реєстраційної форми. Якщо користувача з введеними даними не існує, то введені дані вносяться до бази даних, користувачу призначається роль та на клієнт відправляється інформація про те, що користувача успішно зареєстровано. Якщо користувач з введеними

даними існує, то на клієнт відправляється інформація про те, що користувач з введеними даними існує.

При аутентифікації, спочатку перевіряється наявність в базі даних користувача, у якого електронна адреса співпадає з введеною. Після цього перевіряються паролі – в базі даних та введений. Якщо паролі співпадають, користувач успішно авторизується. Якщо паролі не співпадають, або якщо користувач з введеними даними не існує, на клієнт відправляється інформація про це.

3.3 Опис програмної реалізації клієнтської частини

Клієнтська частина веб-додатку створена за допомогою мови JavaScript та фреймворку React, що дає змогу швидко створювати легко масштабуєму програмну систему.

На рисунку 3.3.1 зображено структуру проекту

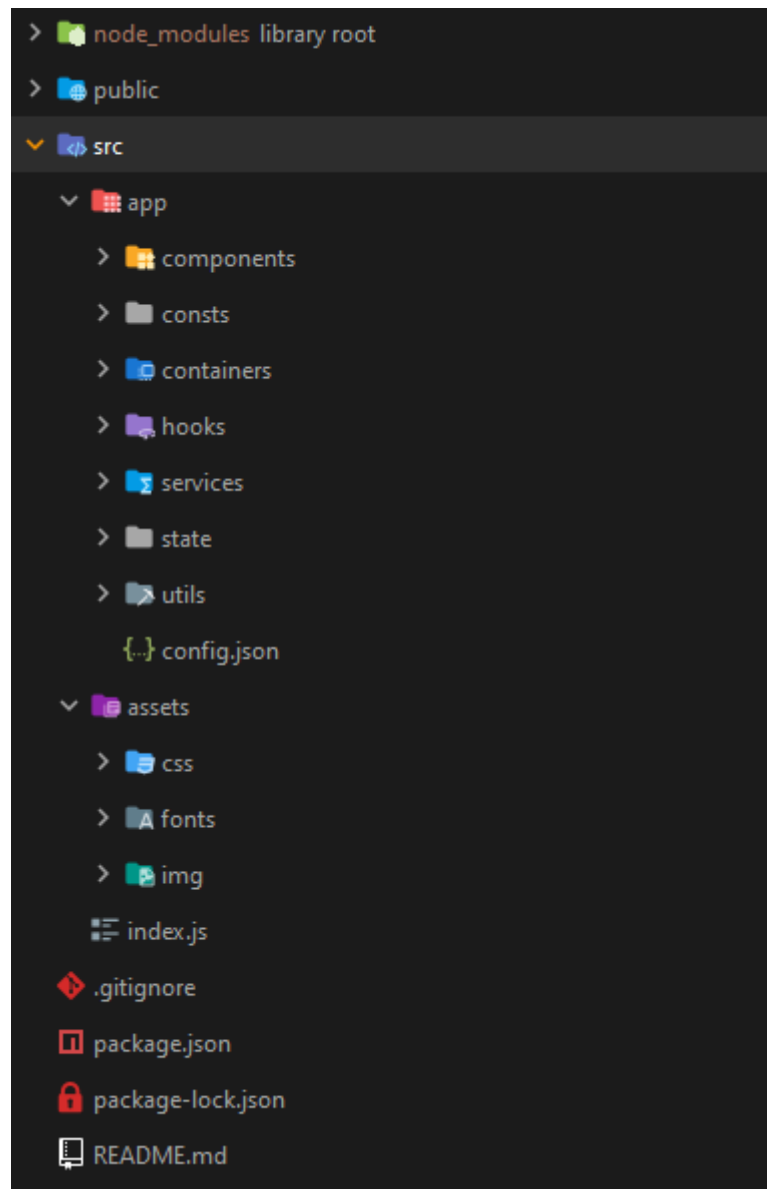


Рисунок 3.3.1 Структура проекту

- **node_modules** – папка, де храняться бібліотеки, необхідні для роботи додатку;
- **public** – папка, де храняться статичні файли (index.html, favicon.ico, тощо);
- **src** – папка з вихідними, «самописними» файлами;
- **src/app** – папка, де храняться файли додатку;
- **src/assets** – папка, де храняться шрифти, стилі, зображення.

3.3.1 Реактивне програмування

Реактивне програмування – це парадигма програмування, побудована на потоках даних та розповсюдженні змін. Тобто, у мовах програмування має бути можливість легко виразити статичні чи динамічні потоки даних, а модель виконання буде автоматично розсилати зміни через потік даних.

Реактивне програмування будується на наступних концепціях:

- **Статика проти динаміки.** Реактивне програмування може бути чисто статичним (потоки даних встановлюються статично), або динамічними (потоки змінюються під час виконання програми).
- **Реактивне програмування вищого порядку.** Реактивне програмування можна назвати реактивним програмуванням вищого порядку, якщо воно підтримує ідею про те, що потоки даних можуть бути використані для побудови інших потоків даних. Це означає, що результуюче значення з потоку даних є інший потік даних, який виконується за тою ж самою моделлю обчислення, що й перший.
- **Диференціація потоку даних.** В ідеальному випадку всі зміни даних поширюються миттєво, але це не може бути забезпечено на практиці. Як вихід, можна задати різні пріоритети обчислення різним частинам графу потоку даних. Це можна назвати диференційованим реактивним програмуванням.
- **Моделі обчислення у реактивному програмуванні.** Обчислення у реактивних програмах не обов'язково засновані на тому ж, що є обчислення у програмах, що будуються на стеках. Замість цього, коли якісь дані змінюються, то зміна поширюється на всі дані, які залежали від змінених даних.
- **Схожість з патерном «Спостерігач».** Реактивне програмування має принципову подібність до патерну «Спостерігач», який зазвичай використовується в ООП. Проте, інтеграція концепцій потоку даних в мову програмування може збільшити ступінь деталізації графа потоку даних.

3.3.2 Взаємодія з серверною частиною.

Для взаємодії клієнтської частини програмної системи з серверною частиною програмної системи, на боці клієнтської частини було реалізовано сервіс (`apiRequestService.js`), який за допомогою AJAX-запитів типу GET або POST передає та отримує дані з серверної частини програмної системи. Сервіс має наступний функціонал:

- **getAllUsers** – **GET** – отримати дані про всіх користувачів;
- **registerUser** – **POST** – зареєструвати користувача;
- **auth** – **POST** – аутентифікувати користувача;
- **getUserDataById** – **POST** – отримати інформацію про користувача за його ідентифікатором;
- **getAllGroups** – **GET** – отримати дані про всі групи;
- **getAllFaculties** – **GET** – отримати список всіх факультетів;
- **addNewGroup** – **POST** – створити нову групу;
- **getAllTeachers** – **GET** – отримати список всіх викладачів;
- **registerTeacher** – **POST** – зареєструвати нового викладача.

На рисунку 3.3.2 зображено схему взаємодії клієнта та сервера.

REST API Design

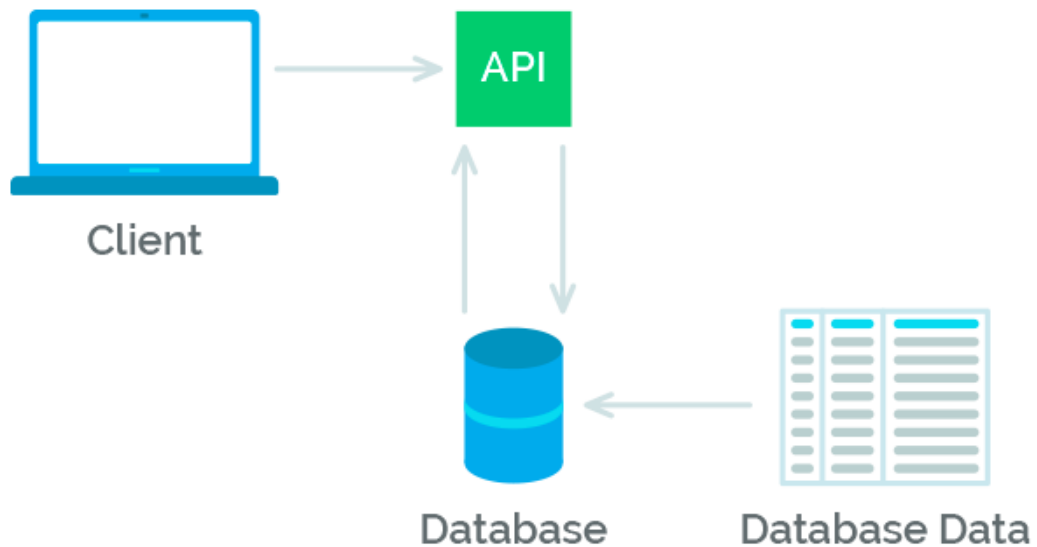


Рисунок 3.3.2 Взаємодія клієнта та сервера

3.3.3 Сторінка аутентифікації або реєстрації користувача

На рисунку 3.3.3 зображено структуру компоненту сторінки аутентифікації або реєстрації користувача.

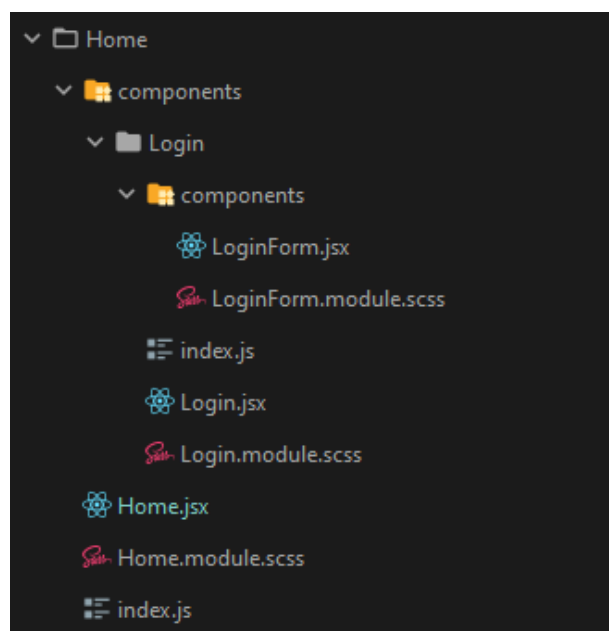


Рисунок 3.3.3 структура компонента сторінки аутентифікації\реєстрації

Сторінка аутентифікації або реєстрації користувача, фактично, є головною сторінкою та єдиною точкою входу користувача до програмної системи. Зі сторінки аутентифікації можна перейти на сторінку подачі реєстраційної форми, сторінку рейтингового списку.

На сторінці аутентифікації або реєстрації користувача користувач може здійснити аутентифікацію у системі для подальшого використання її функціоналом.

3.3.4 Сторінка подання реєстраційної форми

Сторінка подання реєстраційної форми призначена для подання вступником всієї необхідної інформації для його зарахування на третій рівень вищої освіти.

На рисунку 3.3.5 зображено структуру компоненту сторінки подання реєстраційної форми.

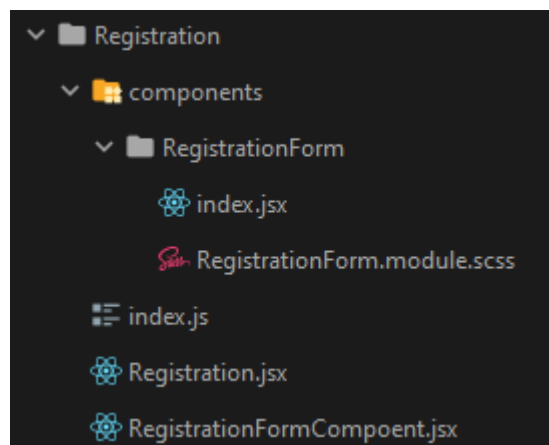


Рисунок 3.3.5 Структура сторінки подання реєстраційної форми

Реєстраційна форма заповнюється в три етапи (детальніше – в розділі 4 Інтерфейс користувача). Перейти до наступного етапу можна тільки після проходження валідації активним етапом. Реєстраційна форма потребує внесення наступних даних:

- прізвище;
- ім'я;
- по батькові;

- статъ;
- дата народження;
- спеціальність – випадуючий список;
- факультет – випадуючий список;
- кафедра – випадуючий список;
- форма навчання – випадуючий список;
- форма оплати – випадуючий список;
- іноземна мова – випадуючий список;
- ВНЗ, що закінчив вступник;
- рік закінчення ВНЗ;
- освітній ступінь – випадуючий список;
- наявність відзнаки;
- електронна пошта;
- пароль;
- потреба в гуртожитку;
- середній бал диплому;
- кількість наукових публікацій;
- прізвище наукового керівника;
- додатковий коментар;

В разі, коли всі поля форми заповнені коректно, по натиску на кнопку відправити, всі дані з реєстраційної форми будуть відправлені за допомогою POST-запиту на серверну частину програмної системи.

4 ІНТЕРФЕЙС ВЗАЄМОДІЇ ТА РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ

Розроблена система є доступною для кожного користувача, який має веб-браузер на комп'ютері або телефоні. Для запуску системи необхідно перейти за посиланням.

4.1 Інтерфейс системи

На головній сторінці системи (рисунок 4.1, рисунок 4.2) користувач в ролі абітурієнта може подати реєстраційну форму або перевірити рейтинговий бал. Адміністратор може авторизуватись в системі для подальшої роботи з даними вступників.

Аспірантура КПІ

Для абітурієнтів

Подати реєстраційну форму

Переглянути рейтинг

Вхід

Email
admin@site.com

Пароль

Увійти

КПІ 2019

Рисунок 4.1 Головна сторінка системи

The image shows a mobile application interface for 'Аспірантура КПІ' (Aspirants of the Kyiv Polytechnic Institute). The interface is displayed on a smartphone screen. At the top, there is a blue header bar containing the Kyiv Polytechnic Institute logo on the left, the text 'Аспірантура КПІ' in the center, and a user profile icon on the right. Below the header, the main content area is white. It features two primary sections: a registration section for applicants and a login section. The registration section, titled 'Для абітурієнтів' (For applicants), contains two blue buttons: 'Подати реєстраційну форму' (Submit registration form) and 'Переглянути рейтинг' (View rating). The login section, titled 'Вхід' (Login), includes two input fields: 'Email' with the text 'admin@site.com' and 'Пароль' (Password) with masked characters '.....'. A blue 'Увійти' (Login) button is positioned below the password field. At the bottom of the screen, a blue footer bar displays the text 'КПІ 2019'.

Аспірантура КПІ

Для абітурієнтів

Подати реєстраційну форму

Переглянути рейтинг

Вхід

Email

admin@site.com

Пароль

.....

Увійти

КПІ 2019

Рисунок 4.2 Головна сторінка системи на смартфоні

4.1.1 Функціональні можливості користувача

Сторінка подачі реєстраційної форми складається з трьох етапів.

На першому етапі користувач вводить свої дані (рисунок 4.3, рисунок 4.4) :

- прізвище;
- ім'я;
- по батькові;
- стать;
- дата народження.

Регістраційна форма абітурієнта

1 Введіть своє ПІБ

Прізвище: Строган Ім'я: Роман По батькові: Сергійович Стать: Чоловіча 04.03.1999

Далі

2 Оберіть спеціальність, факультет і кафедру

3 Вкажіть додаткову інформацію

КПІ 2019

Рисунок 4.3 Перший етап реєстраційної форми

Аспірантура КПІ

Реєстраційна форма абітурієнта

1 Введіть своє ПІБ

Прізвище
Строган

Ім'я
Роман

По батькові
Сергійович

Стать
Чоловіча


04.03.1999

Далі

КПІ 2019

Рисунок 4.4 Перший етап реєстраційної форми на смартфоні

На другому етапі вступник обирає спеціальність, факультет та кафедру (рисунок 4.5, рисунок 4.6).


Аспірантура КПІ


Реєстраційна форма абітурієнта

1 Введіть своє ПІБ

2 **Оберіть спеціальність, факультет і кафедру**

Спеціальність

32 - Історія та архео...

Факультет

ВПІ

Кафедра

ВСР, Кафедра вида...



Назад

Далі

3 Вкажіть додаткову інформацію

КПІ 2019

Рисунок 4.5 Другий етап заповнення реєстраційної форми


Аспірантура КПІ


Реєстраційна форма абітурієнта

1 Введіть своє ПІБ

2 **Оберіть спеціальність, факультет і кафедру**

Спеціальність

32 - Історія та архео...

Факультет

ВПІ

Кафедра

ВСР, Кафедра вида...

Назад

Далі

3 Вкажіть додаткову інформацію

КПІ 2019

Рисунок 4.6 Другий етап заповнення реєстраційної форми на смартфоні

Третій етап подачі реєстраційної форми потребує надання таких даних (рисунок 4.7, рисунок 4.8):

- форма навчання;
- форма оплати;
- іноземна мова;
- ВНЗ який закінчив абітурієнт;
- рік випуску;
- освітній ступінь;
- диплом з відзнакою;
- електронна пошта;
- телефон;
- чи потребує місця в гуртожитку;
- середній бал диплому за 100-бальною шкалою;
- кількість публікацій;
- прізвище передбачуваного наукового керівника;
- особливі відзнаки (перемоги на олімпіадах, патенти, тощо);
- додаткова інформація.

Аспірантура КПІ

Оберіть спеціальність, факультет і кафедру

3 Вкажіть додаткову інформацію

Форма навчання: Вечірня

Форма оплати: Контракт

Іноземна мова: Німецька

Який ВНЗ закінчив: КПІ

Рік випуску: 2004

Спеціаліст/Магістр: Магістр

3 відзнакою

Email

Телефон

Це поле обов'язкове

☒ Потребує місце проживання в гуртожитку на період навчання

Кількість публікацій: 0

Середній бал диплому...

Прізвище передбачуваного наукового керівника

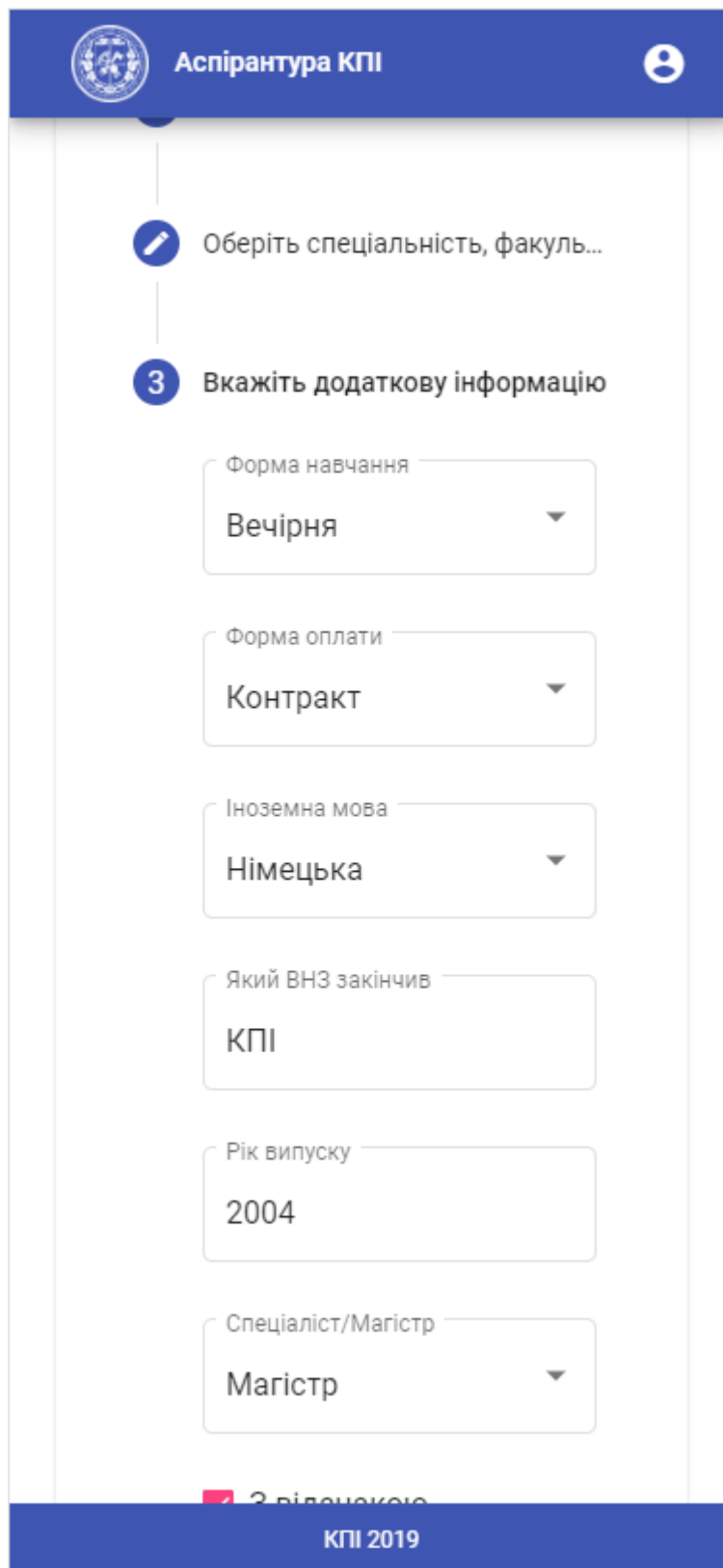
Особливі відзнаки (перемоги на олімпіадах, патенти, тощо)

Додаткова інформація

Назад Завершити реєстрацію

КПІ 2019

Рисунок 4.7 Третій етап заповнення реєстраційної форми



The screenshot shows a mobile application interface for the registration process. At the top, there is a blue header bar with the text "Аспірантура КПІ" (Aspirants of KPI) and a user profile icon. Below the header, a vertical line indicates the progress, with a blue circle containing the number "3" highlighting the current step: "Вкажіть додаткову інформацію" (Specify additional information). The form consists of several input fields with dropdown menus:

- Форма навчання** (Form of study): Set to "Вечірня" (Evening).
- Форма оплати** (Form of payment): Set to "Контракт" (Contract).
- Іноземна мова** (Foreign language): Set to "Німецька" (German).
- Який ВНЗ закінчив** (Which university did you graduate from): Set to "КПІ" (KPI).
- Рік випуску** (Year of graduation): Set to "2004".
- Спеціаліст/Магістр** (Specialist/Master): Set to "Магістр" (Master).

At the bottom of the screen, there is a blue bar with the text "КПІ 2019".

Рисунок 4.8 Третій етап заповнення реєстраційної форми на смартфоні

Після успішної реєстрації, вступник може перевірити свій рейтинговий бал за

допомогою форми пошуку абітурієнтів (рисунок 4.9)

ПІБ	Факультет	Спеціальність	Рейтинговий бал
Матвієнко Андрій Миколайович	ІПСА	111	15.315000000000001

Рисунок 4.9 Форма пошуку вступників

Вступник має змогу переглянути список вступників за спеціальністю та визначити своє місце в рейтингу (рисунок 4.10).

Рейтинговий список вступників за спеціальністю 111 (Математика).
Форма навчання: Денна.
Кількість бюджетних місць: 3

№	ПІБ	Факультет	Спеціальність	Форма навчання	Рейтинговий бал
1	Матвієнко Андрій Миколайович	ІПСА	111	Денна	15.315000000000001
2	Магула Стас Григорович	ІПСА	111	Денна	13.93
3	Малиновський Андрій Михайлович	ІПСА	111	Денна	13.804999999999998
4	Пригожина Анастасія Юріївна	ІПСА	111	Денна	13.07

Рисунок 4.10 Рейтинговий список вступників

4.1.2 Функціональні можливості адміністратор

В разі успішної авторизації, адміністратор системи має можливість переглянути список поданих заяв (рисунок 4.11). В таблиці можна побачити перелік стислих даних про вступника, та наявна кнопка, яка веде до детальної сторінки поданої заяви.

Аспірантура КПІ

Вітаємо, Дмитро Врадій

Подані заяви

Абітурієнти

Зараховані

Статистика

Пошук

ПІБ	Факультет	Спеціальність	
Северин Микита Андрійович	ФММ	75	<div>Детальніше</div>
Шишута Анатолій Миколайович	ТЕФ	144	<div>Детальніше</div>
Козловський Андрій Геннадійович	ММІ	131	<div>Детальніше</div>
Борщик Сергій Олександрович	ММІ	133	<div>Детальніше</div>
Солончук Ірина Вікторівна	ФСП	81	<div>Детальніше</div>
Котковський Богдан Сергійович	ФБМІ	163	<div>Детальніше</div>
Романенко Ірина Ігорівна	ТЕФ	143	<div>Детальніше</div>
Щоголева Світлана Геннадіївна	ММІ	173	<div>Детальніше</div>
Мощенко Ілля Михайлович	ХТФ	161	<div>Детальніше</div>
Назаренко Олена Володимирівна	ВПІ	186	<div>Детальніше</div>
Гумінський Артем Миколайович	ТЕФ	122	<div>Детальніше</div>
Бойко Дмитро Сергійович	ТЕФ	122	<div>Детальніше</div>
Євтушенко Артем Михайлович	ТЕФ	121	<div>Детальніше</div>

localhost:4200/cabinet/registration-data

КПІ 2019

Рисунок 4.11 Список поданих заяв

На детальній сторінці поданої заяви (рисунок 4.12), адміністратор може переглянути дані вступника та прийняти рішення щодо підтвердження або скасування реєстраційної заяви вступника.

Якщо адміністратор відмовляє в реєстрації, реєстраційна заява вступника видаляється з системи. Після підтвердження реєстрації картка вступника переходить до списку абітурієнтів.

The screenshot shows the 'Aspirantura KPI' interface. On the left is a sidebar with navigation links: 'Подані заяви', 'Абітурієнти' (highlighted), 'Зараховані', and 'Статистика'. The main area has two buttons at the top: 'Підтвердити реєстрацію' (blue) and 'Відмовити в реєстрації' (red). Below these is a table with the following data:

Абітурієнт:	Северин Микита Андрійович
Стать:	Чоловіча
Дата народження:	15 лип. 1994 р.
Факультет:	ФММ
Кафедра:	ПМФММ
Спеціальність:	75
Форма навчання:	Вечірня
Форма оплати:	Держзамовлення
Іноземна мова:	Англійська
ВНЗ:	КПІ
Рік випуску:	2018
Диплом:	Магістр з відзнакою

Рисунок 4.12 Детальна сторінка заяви абітурієнта

На сторінці «Абітурієнти» адміністратору доступний до перегляду список вступників з підтвердженою реєстрацією (рисунок 4.13)

The screenshot shows the 'Aspirantura KPI' interface with the 'Абітурієнти' section selected. A search bar labeled 'Пошук' is at the top. Below it is a table with the following data:

ПІБ	Факультет	Спеціальність	
Матвієнко Андрій Миколайович	ІПСА	111	Детальніше
Магула Стас Григорович	ІПСА	111	Детальніше
Пригожина Анастасія Юрївна	ІПСА	111	Детальніше
Малиновський Андрій Михайлович	ІПСА	111	Детальніше

Рисунок 4.13 Список вступників з підтвердженою реєстрацією.

На детальній сторінці (рисунок 4.14), адміністратор може переглянути дані вступника, а також занести бали за вступні екзамени до системи. Система автоматично згенерує рейтинговий бал та розмістить вступника у рейтинговому списку.

Аспірантура КПІ

Вітаємо, Дмитро Врадій

Подані заяви

Абітурієнти

Зараховані

Статистика

Абітурієнт: Матвієнко Андрій Миколайович

Екзамен з іноземної мови

9

Фаховий екзамен

21

Додаткові бали

1

☐ Потребує додаткового іспиту

Додатковий вступний іспит

Рейтинговий бал: 15.315000000000001

Зберегти зміни

Зарахувати абітурієнта

Видалити з бази

Стать:	Чоловіча
Дата народження:	10 лип. 1997 р.
Факультет:	ІПСА
Кафедра:	ММСА
Спеціальність:	111

Рисунок 4.14 Детальна сторінка абітурієнта

На сторінці «Зараховані» відображаються всі зараховані вступники (рисунк 4.15), адміністратор має змогу присвоїти аспіранту групу (рисунк 4.16).

Аспірантура КПІ

Вітаємо, Дмитро Врадій

Подані заяви

Абітурієнти

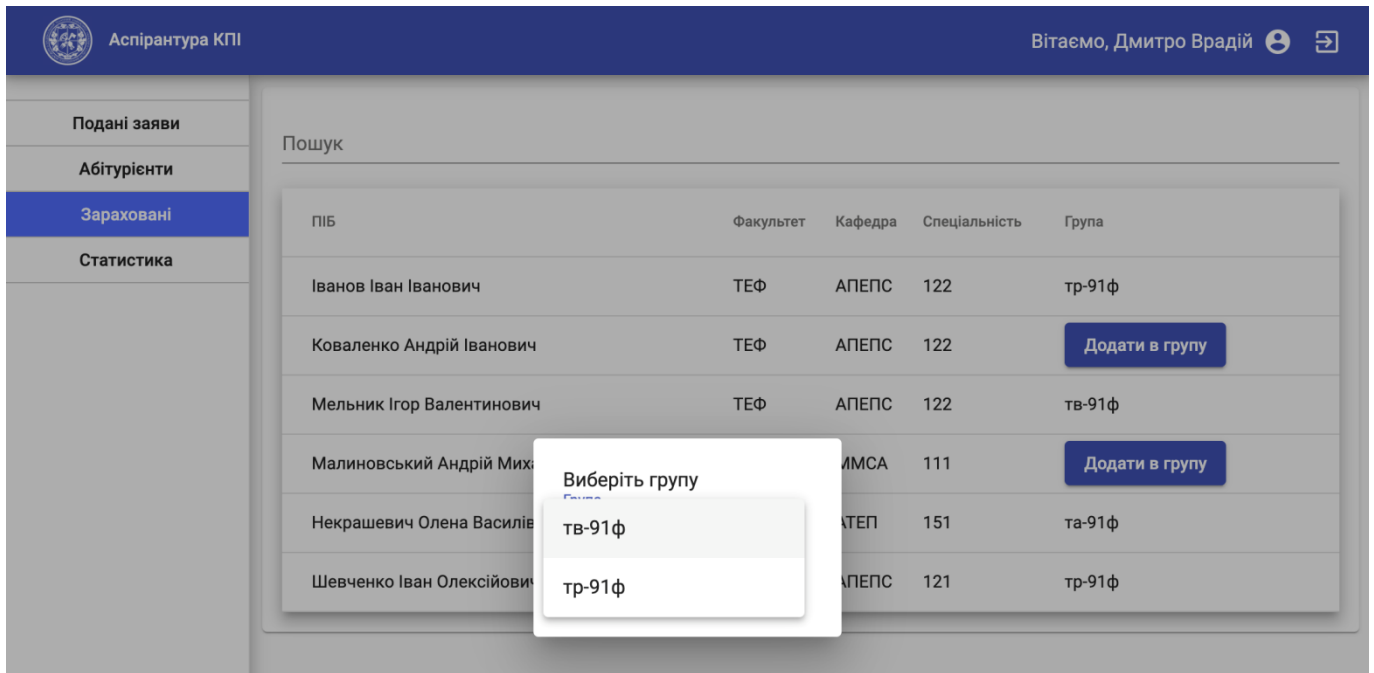
Зараховані

Статистика

Пошук

ПІБ	Факультет	Кафедра	Спеціальність	Група
Іванов Іван Іванович	ТЕФ	АПЕПС	122	тр-91ф
Мельник Ігор Валентинович	ТЕФ	АПЕПС	122	тв-91ф
Некрасевич Олена Василівна	ТЕФ	АТЕП	151	та-91ф
Шевченко Іван Олексійович	ТЕФ	АПЕПС	121	тр-91ф

Рисунок 4.15 Список зарахованих вступників



Аспірантура КПІ

Вітаємо, Дмитро Вradій

Пошук

ПІБ	Факультет	Кафедра	Спеціальність	Група
Іванов Іван Іванович	ТЕФ	АПЕПС	122	тр-91ф
Коваленко Андрій Іванович	ТЕФ	АПЕПС	122	Додати в групу
Мельник Ігор Валентинович	ТЕФ	АПЕПС	122	тв-91ф
Малиновський Андрій Михайлович	ТЕФ	АПЕПС	122	Додати в групу
Некрашєвич Олена Василівна	ТЕФ	АПЕПС	122	тв-91ф
Шевченко Іван Олексійович	ТЕФ	АПЕПС	122	тв-91ф

Виберіть групу

- тв-91ф
- тр-91ф

4.16 Присвоєння зарахованому вступнику групи

Адміністратор системи може переглядати статистику вступної компанії по різним критеріям (рисунок 4.17):

- а) факультет – подана заяві
- б) факультет – абітурієнти
- в) факультет - зараховані

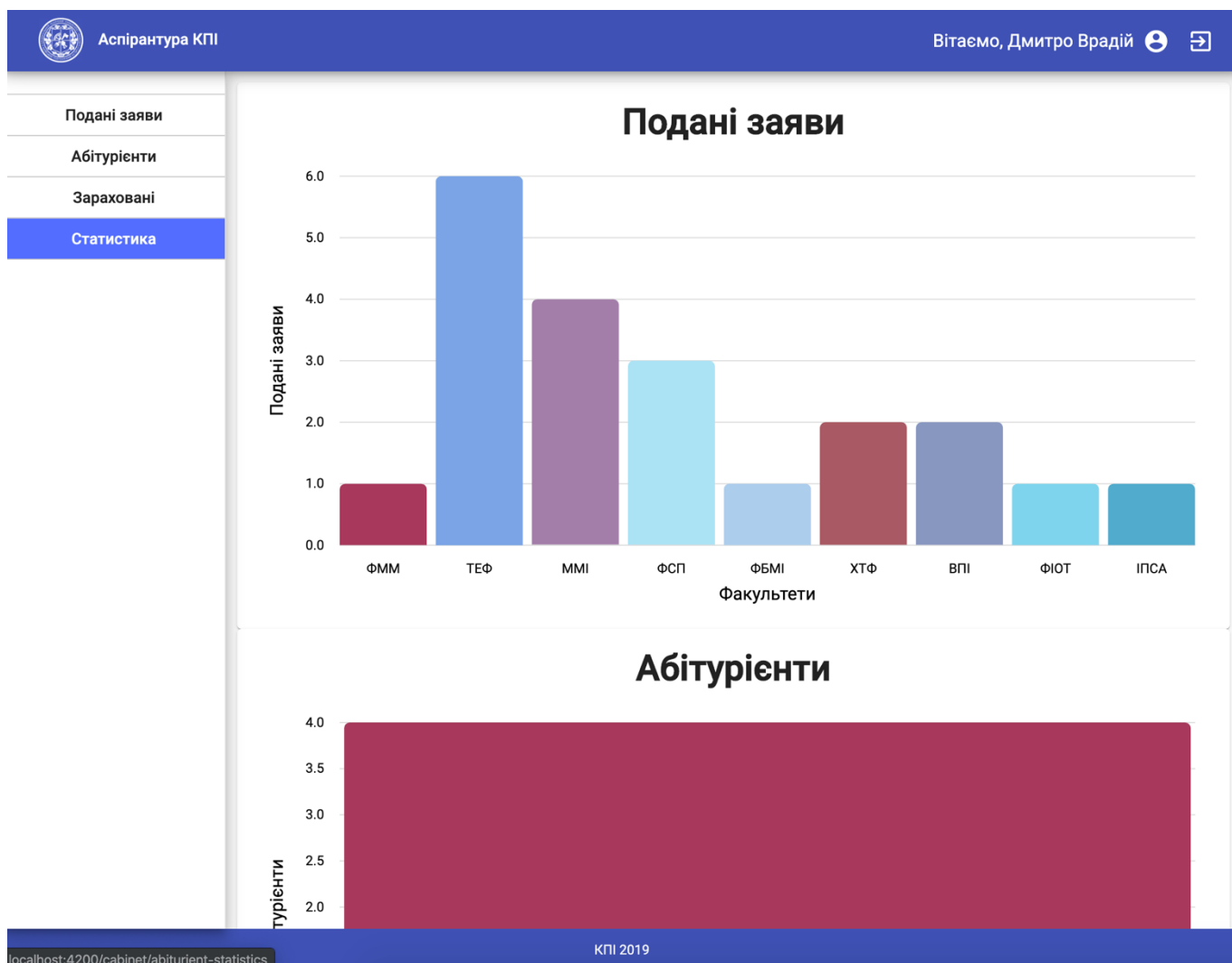


Рисунок 4.17 – Статистика вступної кампанії

4.2 Висновки до розділу

У цьому розділі наведено опис інтерфейсу програмної системи, яка дає змогу проводити супроводження, моніторинг та аналіз вступної кампанії до аспірантури у вищому навчальному закладі.

Програмне забезпечення надає можливість абітурієнту подати реєстраційну форму та брати участь у вступній кампанії. Вступник, після реєстрації, може переглянути свій вступний бал.

Адміністратор системи може проводити всі етапи вступної кампанії. Він може переглядати реєстраційні заявки вступників та підтверджувати або скасовувати їх. Для підтверджених заяв можна внести бали за вступні экзамены, присвоїти їм групу.

Розділ статистики дозволяє адміністратору аналізувати проходження вступної кампанії.

У ході тестування додатку не було помічено помилок або порушень в роботі, тож додаток можна використовувати на приладах, які підтримують роботу веб-браузерів.

ВИСНОВКИ

В результаті дослідження було розроблено програму систему супроводження, моніторингу та аналізу вступу на навчання до аспірантури Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Систему було розроблено у вигляді веб-додатку. Це сприяло покращенню та вдосконаленню навичок розробки програмного забезпечення, а також покращенню володіння мовою програмування JavaScript. Також, були отримані навички роботи з платформою Node.js та системою керування базами даних MongoDB.

Завдяки використанню мови програмування JavaScript та фреймворку React.js на клієнтській частині та платформи Node.js та системи керування базами даних MongoDB на серверній частині програмної системи, вдалося досягти поставлених цілей та розробити програмну систему, яка може легко масштабуватись, підтримуватись а також є дружньою до підключення нових модулів.

Завдяки розробці програми у вигляді веб-застосунку, її доступність є практично максимальною для усіх користувачів.

Таким чином, розроблена система дає змогу брати участь у вступній кампанії на третій рівень освіти всім абітурієнтам та контролювати процес приймальною комісією. Контроль процесу та всі необхідні дії здійснюються в одному місці, що дає змогу автоматизувати процеси, зменшити кількість помилок через людський фактор.

Розроблена програмна система є крос-платформною, може використовуватись на великій кількості платформ. Для запуску та роботи системи необхідно використовувати сучасний веб-браузер, який присутній на багатьох пристроях.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Міністерство освіти і науки України. Умови прийому на навчання до закладів вищої освіти України в 2019 році/ Міністерство освіти і науки України — Київ, 2019 — 54 с.
2. Флэнаган Д. JavaScript. Подробное руководство / Дэвид Флэнаган — Символ-Плюс 2013. — 1080 с.
3. Стефанов С. JavaScript. Шаблоны проектирования / Стефанов Стоян. — Санкт-Петербург: Лори, 2011. — 272 с.
4. Хэмптон С. CSS3 для профессионалов / Сэм Хэмптон., 2016. — 183 с
5. БенКристенсен — «Reactive Programming with RxJava: Creating Asynchronous», 2014 р. — 560 с.